A. Requirements

Code (90%)

You can write your code in Java, Python, C, or C++. The *time limit* may vary among different languages, depending on the performance of the language. Your code must be a complete excutable program instead of only a function. We guarantee test data strictly compliance with the requirements in the description, and you do not need to deal with cases where the input data is invalid.

Libraries in this assignment:

- For C/C++, you can only include standard library.
- For Java, you can only `import java.util.*`
- For Python, you can only import standard library. In other words, you cannot import libraries such as `numpy`.

We provide an example problem to illustrate the information above better.

Report (10%)

You also need to write a report in `pdf` type to explain the following:

- What are the possible solutions for the problem?
- How do you solve this problem?
- Why is your solution better than others?

Please note that the **maximum** number of pages allowed for your report is **5 pages**.

Remember that the report is to illustrate your thinking process. Keep in mind that your report is supposed to show your ideas and thinking process. We expect clear and precise textual descriptions in your report, and we do not recommend that you over-format your report.

B. Example Problem: A + B Problem

Description

Given 2 integers A and B, compute and print A + B

Input

Two integers in one line: A, and B

Output

One integer: A + B

Sample Input 1

```
1 2
```

Sample Output 1

```
3
```

Problem Scale & Subtasks

For 100% of the test cases, $0 \leq A, B \leq 10^6$

Solutions

Java

```java
import java.util.*;

public class Example {
    public static void main(String[] args) {
        int a, b;
        Scanner scanner = new Scanner(System.in);
        a = scanner.nextInt();
        b = scanner.nextInt();
        scanner.close();
        System.out.println(a + b);
    }
}
```

Python

```python
AB = input().split()
A, B = int(AB[0]), int(AB[1])
print(A + B)
```

C

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  int A, B;
  scanf("%d%d", &A, &B);
  printf("%d\n", A + B);
  return 0;
}
```

C++

```cpp
#include <iostream>

int main(int argc, char *argv[])
{
  int A, B;
  std::cin>> A >> B;
  std::cout<< A + B << std::endl;
  return 0;
}
```

C. Submission

After finishing this assignment, you are required to submit your code to the Online Judge System (OJ), and upload your `.zip` package of your code files and report to BlackBoard.

C.1 Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (oj.cuhk.edu.cn, campus only) to gain marks for the code part. You can submit your solution of one problem for **no more than 80 times**.

After you have submitted your program, OJ will test your program on all test cases and give you a grade. The grade of your latest submission will be regarded as the final grade of the corresponding problem. Each problem is tested on multiple test cases of different difficulty. You will get a part of the score even if your algorithm is not the best.

Note: The program running time may vary on different machines. Please refer to the result of the online judge system. OJ will show the time and memory limits for different languages on the corresponding problem page.

If you have other questions about the online judge system, please refer to OJ wiki (campus network only). If this cannot help you, feel free to contact us.

C.2 BlackBoard

You are required to upload your **source codes and report** to the BlackBoard platform. You need to name your files according to the following rules and compress them into `A1_<Student ID>.zip` :

```
A1_<Student ID>.zip
|-- A1_P1_<Student ID>.java/py/c/cpp
|-- A1_P2_<Student ID>.java/py/c/cpp
|-- A1_Report_<Student ID>.pdf
```

For Java users, **you don't need to consider the consistency of class name and file name**.

For example, suppose your ID is 123456789, and your problem 1 is written in `Python`, problem 2 is written in `Java` then the following contents should be included in your submitted `A1_123456789.zip`:

```
A1_123456789.zip
|-- A1_P1_123456789.py
|-- A1_P2_123456789.java
|-- A1_Report_123456789.pdf
```

C.3 Late Submissions

Submissions after Nov 24 2023 23:59:00(UTC+8) would be considered as LATE.

The LATE submission page will open after deadline on OJ.

Submisson time = max{latest submisson time for every problem, BlackBoard submisson time}

There will be penalties for late submission:

- 0–24 hours after deadline: final score = your score×0.8
- 24–72 hours after deadline: final score = your score×0.5
- 72+ hours after deadline: final score = your score×0

FAQs

Q: I cannot access to Online Judge.

A: First, please ensure that you are using the campus network. If you are not on campus, please use the university VPN. Second, please delete cookies and refresh browser or use other browser. If you still cannot access to Online Judge, try to visit it via the IP address `10.26.200.13`.

Q: My program passes samples on my computer, but not get AC on OJ.

A: Refer to OJ Wiki Q&A

Authors

If you have questions for the problems below, please contact:

- Yige Jiang: 121090233@link.cuhk.edu.cn
- Ruiying Liu: ruiyingliu@link.cuhk.edu.cn

CSC3100 Data Structures Fall 2023

Programming Assignment 3

Due: Nov 24 2023 23:59:00

Assignment Link: http://oj.cuhk.edu.cn/contest/csc310023falla3

Access Code: `9v7Dxqet`

1 Node Distance(40% of this assignment)

1.1 Description

You are given a tree with n nodes, where each edge in the tree has a corresponding weight denoting the length of each edge. The nodes in the tree are colored either black or white. Your task is to calculate the sum of distances between every pair of black nodes in the tree. Let $B = \{b_1, b_2, ...\}$ a set of black nodes, then the answer is formulated as:

$$Ans = \sum_{i=1}^{|B|-1} \sum_{j=i+1}^{|B|} \text{dist}(b_i, b_j)$$

where $|B|$ denotes the number of the black nodes in the tree, and $\text{dist}(b_i, b_j)$ is the length of the simple path from the i-th to j-th black node.

Write a program to calculate the sum of distances on the tree between every pair of black nodes Ans in the given tree.

1.2 Input

The first line contains an integer n, representing the number of nodes in the tree.

The second line contains n space-separated integers $\{c_1, c_2, \ldots, c_i, \ldots, c_n\}$ where c_i is either 0 or 1. $c_i = 1$ indicates that the i-th node is black, and $c_i = 0$ indicates that the i-th node is white.

The following $n-1$ lines, $\{l_1, l_2, \ldots, l_p, \ldots, l_{n-1}\}$, denoting the structure of the tree follow, each line l_p contains 2 integers q_p and w_p, denoting an edge of length w_p between the $p+1$-th node and the q_p-th node.

1.3 Output

Output the sum of distances for every pair of black nodes in the tree.

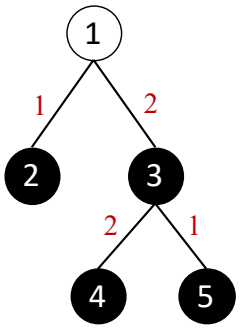
Sample Input 1

```
5
0 1 1 1 1
1 1
1 2
3 2
3 1
```

Sample Output 1

```
18
```

This sample considers a tree with 5 nodes:



The 1-st node is white, and 2-, 3-, 4-, 5-th nodes are black.
The length of edge: (2-nd, 1-st): 1, (3-rd, 1-st): 2, (4-th, 3-rd): 2, (5-th, 3-rd): 1.
$Ans = ((1+2)+(1+2+2)+(1+2+1))+(2+1)+2+1 = 18$.

Sample Input 2

```
9
0 1 0 1 1 1 1 1 1
1 2
1 3
2 2
2 1
5 2
5 3
1 2
7 1
```

Sample Output 2

```
96
```

Three additional large-scale samples are included in the provided files, namely, `A_samplecase1.in/.ans`, `A_samplecase2.in/.ans` and `A_samplecase3.in/.ans`.

Problem Scale & Subtasks

For 100% of the test cases, $1 \leq n \leq 10^5$, $1 \leq q_{p-1} < p$, $1 \leq w_p \leq 1000$

Test Case No.	Constraints
1-4	$n \leq 100$
5-7	$n \leq 1000$
8	$q_p = p$
9	$q_p = 1$
10	No additional constraints

Hint

It can be proven that the given structure is definitely an unrooted tree.

For C/C++ and Java users, an `int` type stores integers range from -2,147,483,648 to 2,147,483,647. It may be too small for this problem. You need other data types, such as `long long` for C/C++ and `long` for Java. They store integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Use `scanf("%lld",&n)` for C, `cin>>n` for C++ and `n = scanner.nextLong()` for Java to get the input n. And the other operations for `long` and `long long` are quite same as `int`.

For Python users, if there occurs a `RecusrionError`, see here.

2 Price Sequence (50% of this assignment)

2.1 Description

Mario bought n math books and he recorded their prices. The prices are all integers, and the price sequence is $a = \{a_0, a_2, ...a_i, ..., a_{n-1}\}$ of length n ($n \leq 100000$). Please help him to manage this price sequence. There are three types of operations:

- `BUY x`: buy a new book with price x, thus x is added at the end of a.
- `CLOSEST_ADJ_PRICE`: output the minimum absolute difference between adjacent prices.
- `CLOSEST_PRICE`: output the absolute difference between the two closest prices in the entire sequence.

A total of m operations are performed ($1 \leq m \leq 100000$). Each operation is one of the three mentioned types. You need to write a program to perform given operations. For operations "`CLOSEST_ADJ_PRICE`" and "`CLOSEST_PRICE`" you need to output the corresponding answers.

2.2 Input

The first line contains two integers n and m, representing the length of the original sequence and the number of operations.

The second line consists of n integers, representing the initial sequence a.

Following that are m lines, each containing one operation: either `BUY x`, `CLOSEST_ADJ_PRICE`, or `CLOSEST_PRICE` (without extra spaces or empty lines).

2.3 Output

For each `CLOSEST_ADJ_PRICE` and `CLOSEST_PRICE` command, output one line as the answer.

Sample Input 1

```
3 4
7 1 9
CLOSEST_ADJ_PRICE
BUY 2
CLOSEST_PRICE
CLOSEST_ADJ_PRICE
```

Sample Output 1

```
6
1
6
```

Sample Input 2

```
6 12
30 50 39 25 12 19
BUY 4
CLOSEST_PRICE
BUY 14
CLOSEST_ADJ_PRICE
CLOSEST_PRICE
BUY 0
CLOSEST_PRICE
BUY 30
BUY 12
CLOSEST_PRICE
BUY 20
CLOSEST_PRICE
```

Sample Output 2

```
5
7
2
2
0
0
```

Two additional large-scale samples are included in the provided files, namely, `B_samplecase1.in/.ans` and `B_samplecase2.in/.ans`.

Problem Scale & Subtasks

For 100% of the test cases, $2 \leq n, m \leq 1 \times 10^5$, $0 \leq a_i, x \leq 10^{12}$

Test Case No.	Constraints
1-4	$n \leq 10^3, m \leq 10^3$
5-6	There is no `CLOSEST_PRICE` operation
7-9	a_i and x are uniformly distributed at random within the range $[0, 10^{12}]$
10	No additional constraints

Hint

For C/C++ and Java users, an `int` type stores integers range from -2,147,483,648 to 2,147,483,647. It may be too small for this problem. You need other data types, such as `long long` for C/C++ and `long` for Java. They store integers ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Use `scanf("%lld",&n)` for C, `cin>>n` for C++ and `n = scanner.nextLong()` for Java to get the input n. And the other operations for `long` and `long long` are quite same as `int`.

For Python users, if there occurs a `RecusrionError`, see here.